Bank Robbery

You just got word that a bank robber is robbing the NBEAM (National Bank with Enormous Amounts of Money). The police of course wants to catch him, but they can not storm the building since he has taken hostages. Their plan is to catch him on his way out of the country.

Unfortunately, the police only has a very limited amount of personnel, but they want to be absolutely sure to catch that bank robber.

To catch him, they can build roadblocks on a few roads and check every person using those roads. Since some roads are large and some small, they all need different amounts of policemen to be blocked.

At dinner, Lea's father complains that he has been appointed "Head of Strategic Positioning" but he has no idea how to distribute the policemen. Can you tell them if they have enough policemen to cover every possible route out of the country?

Input

The first line of the input contains an integer t. t test cases follow.

Each test case begins with a line containing three integers l, the amount of available policemen, n, the amount of intersections and m, the amount of roads. m lines follow, each consisting of three integers i, j, k, specifying a road from intersection i to j with k being the amount of policemen it takes to construct a roadblock on it. All roads are useable in both directions. The robber always starts at intersection 1 and wants to get to the border (intersection n). Every test case ends with a blank line.

Output

For each test case, print a line containing "Case #i: t" with i being the number of the test case, starting at 1, and t being "yes" if the minimal amount of policemen it takes to construct at least one road block on every path from intersection 1 to n is smaller or equal to l, "no" otherwise.

Constraints

- $1 \leq t \leq 20$
- $1 \leq l \leq 10000$
- $2 \leq n \leq 400$
- $1 \leq m \leq 100000$
- $1 \leq i, j \leq n$
- $1 \leq k \leq 10000$
- There will always be at least one possible path from 1 to n.

Sample Input 1	Sample Output 1
2 10 2 3 2 2 3 1 2 4 1 1 3 2 3 3 1 3 5 1 2 5 3 3 5	Case #1: yes Case #2: no

Sample Input 2

```
6
8 2 3
2 2 4
1 2 5
1 1 4

2 5 5
1 5 1
5 4 1
4 2 1
5 3 1
1 2 1

1 9 11
1 4 1
4 8 1
8 2 1
2 7 1
2 3 1
7 5 1
1 6 1
6 9 1
2 8 1
2 9 1
5 7 1

2 7 9
1 6 1
1 5 1
1 4 1
4 3 1
1 2 1
3 7 1
2 6 1
4 5 1
6 7 1

1 3 3
1 2 1
1 3 1
1 3 1

2 3 2
1 3 1
3 2 1
```

Sample Output 2

```
Case #1: yes
Case #2: yes
Case #3: no
Case #4: yes
Case #5: no
Case #6: yes
```